

B

Requested Patent: JP10320338A

Title:

SYSTEM FOR CONTROLLING UNITARY MANAGEMENT AND PROCESSING  
EXECUTION OF CSS SERVER ;

Abstracted Patent: JP10320338 ;

Publication Date: 1998-12-04 ;

Inventor(s): MINAMI SHOICHI; TOKUNAGA TOSHIAKI ;

Applicant(s): HITACHI LTD ;

Application Number: JP19970128285 19970519 ;

Priority Number(s): ;

IPC Classification: G06F13/00 ;

Equivalents: ;

ABSTRACT:

PROBLEM TO BE SOLVED: To manage the activated states of all servers and the executed state of an application by a server positioned as a managing server, and to control of the execution of an application operating on the other slave server and the managing server from the managing server in a system constituted of a CSS (client server system) configuration.

SOLUTION: In a system constituted of a CSS configuration, a managing server 100 is provided with a communication file 130 for storing the system activating states of all servers. The contents of the communication files 130 and 230 are referred and updated by the transmission and reception of a message between the managing server 100 and a slave server 200, and the activated state of each server is captured and monitored based on information held by the communication file 130. Also, an application execution state in each server is held by the communication files 130 and 230 of each server, and the application execution state held in the communication file is checked so that the validity of the execution of a processing whose execution is requested in each server can be judged, and when the processing is executable, the pertinent processing is executed.

(51)Int.Cl.<sup>6</sup>

G 06 F 13/00

識別記号

3 5 7

F I

G 06 F 13/00

3 5 7 Z

## 審査請求 未請求 請求項の数2 OL (全11頁)

(21)出願番号 特願平9-128285

(22)出願日 平成9年(1997)5月19日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 南 章一

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

(72)発明者 徳永 俊昭

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

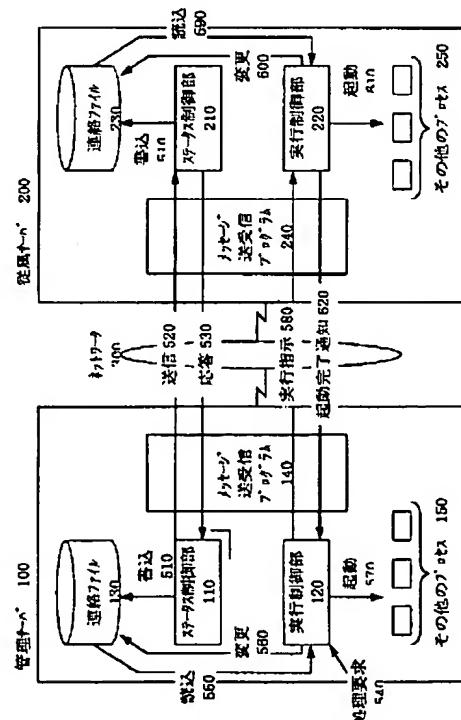
(74)代理人 弁理士 小川 勝男

## (54)【発明の名称】 CSSサーバの一元管理及び処理実行制御方式

## (57)【要約】

【課題】CSS形態で構成されるシステムにおいて、管理サーバとして位置付けられたサーバで、全てのサーバの稼働状態、アプリケーションの実行状態を管理し、かつ管理サーバからその他の従属サーバと管理サーバ上で動作するアプリケーションの実行制御する。

【解決手段】CSS形態で構成されるシステムにおいて、管理サーバ100に、全サーバのシステム稼働状態を格納するための連絡ファイル130を設ける。管理サーバと従属サーバ間のメッセージの送受信により、連絡ファイル130、230の内容を参照・更新し、連絡ファイル130に保有された情報に基づき各サーバの稼働状態の把握・監視を行う。また、各サーバの連絡ファイル130、230に各サーバでのアプリケーション実行状態を保有し、連絡ファイル内に保有しているアプリケーション実行状態をチェックすることにより、各サーバで実行要求のあった処理の実行可否を判断し、実行可能な場合に該当する処理を実行する。



**【特許請求の範囲】**

**【請求項1】**同一ネットワークに接続されているクライアントサーバシステム(CSS)形態で稼働している複数台のサーバ内に、それぞれステータス制御部とアプリケーションの実行制御部を設け、複数台のサーバのうち1台を管理サーバとして位置付け、その他のサーバを従属サーバとして位置付けて、これらの管理サーバと従属サーバの間で必要な情報を伝達するための連絡ファイルを管理サーバと従属サーバにそれぞれ設け、前記のステータス制御部では、予め設定された条件により管理サーバと従属サーバ間でメッセージを送受信することにより各サーバの稼働状態を監視し、その結果を前記の連絡ファイルに出力する。さらに前記アプリケーションの実行制御部では、オンラインやバッチ等のサーバ内アプリケーションの実行処理ステータスのチェックを行い、その結果を前記連絡ファイルに出力し、管理サーバで管理サーバと従属サーバでのアプリケーションの実行処理ステータスの一元管理を可能とし、各サーバの稼働状態とアプリケーション処理ステータスを一元管理することを特徴とするCSSシステムにおけるCSSサーバの一元管理及び処理の実行制御方式。

**【請求項2】**前記連絡ファイルを利用し、一定間隔で管理サーバへ送信される複数台の従属サーバのオンラインやバッチ等のサーバ内アプリケーションの実行処理ステータスを管理サーバで参照を行い、管理サーバのアプリケーションの実行制御部は、オペレータや定刻に起動される管理サーバまたは従属サーバに対する一括処理実行要求に対して、各管理サーバまたは従属サーバの管理サーバ内の連絡ファイルに保有するアプリケーションの実行処理ステータスに応じた実行可能アプリケーション処理を判定し、アプリケーションの実行制御することを特徴とするCSSサーバの一元管理及び処理実行制御方式。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】**本発明は、CSS形態のシステムにおいてサーバ間の実行処理制御に関し、特に稼働状態とアプリケーション実行状態といったステータスを効率良く管理・制御するための一元管理と実行制御方式。

**【0002】**

**【従来の技術】**従来、CSS形態のシステムでは、サーバ・クライアント・ネットワークといった構成要素について一定のポーリング方式をもとに、1台のサーバで稼働状態を管理する方式が採用されている。このような方式の一例として、特開平4-153855号公報に記載されているネットワークリンク一元管理方式がある。

**【0003】**本方式では、複数のホストが複雑に接続されたネットワーク環境において、障害情報を含むネットワークの稼働状況情報をそれぞれのホストにおいて採取

し、それぞれのホストはその情報を特定のホストにその稼働状況情報を送信を行う。その特定のホスト上のネットワーク稼働状況の集中監視と各種ネットワークの操作／状態の一元化を実現し、ネットワーク障害の迅速な把握を目的としている。

**【0004】**しかし、このような方式では、サーバ・クライアント・ネットワークといった構成要素の稼働状態を管理するのみであり、システム全体での整合性のとれた運用制御を実現するには至っていない。その一方でCSSシステムにおいては、大規模分散化が急速に進んでおり、統一的な運用の実現と運用負荷の軽減の為に、システムの稼働状態の一元管理のみでなく、これらの情報をもとに予め設定した実行条件に従い、分散配置されたサーバの運用を制御するための処理実行制御方式が望まれている。

**【0005】**

**【発明が解決しようとする課題】**上記従来の方式では、ネットワークの状態を管理する方式として、ネットワークに障害が発生した時のオペレータの運用負荷軽減と操作の簡略化を狙いとしている。従って、オンラインやバッチまたは非稼働状態に区分されるアプリケーションの実行ステータスに基づく、管理サーバと従属サーバのサーバアプリケーションの実行処理の運用についても、先に述べたネットワークの運用と同様にサーバ通常運用や障害運用といった運用局面毎にオペレータに対して運用負荷がかかると同時にサーバでのコマンド入力等の操作が発生する。その際の問題点としては、以下のようなことがあげられる。

**【0006】**(1) 管理サーバと従属サーバについて、それが接続されているネットワークを含めた、各サーバのOS起動の有無のレベルで考えたサーバの稼働または非稼働といった、サーバの生死監視については、管理サーバにより一元管理可能であるが、管理サーバまたは従属サーバでのオンラインやバッチまたは非稼働のアプリケーションの実行ステータスについては、各サーバで別々に保有しており、管理サーバでの生死監視と同様に、管理サーバと従属サーバのアプリケーションの実行処理ステータスの一元管理は不可能であるという問題点がある。

**【0007】**(2) 管理サーバや従属サーバでそれぞれ個別にオンラインやバッチまたは非稼働のアプリケーションの実行処理ステータスを保有管理しているため、管理サーバでは自身のアプリケーションの実行処理ステータスは保有管理しているが、従属サーバのアプリケーションの実行管理ステータスは保有管理していないため、各従属サーバのアプリケーションの実行処理ステータスを意識した形での処理アプリケーションの実行要求を一括してアプリケーションの実行することはできない。従って、アプリケーションの実行処理ステータスを管理サーバと従属サーバで保有管理していても、運用の局面に

応じた処理アプリケーションの実行要求は人間系の伝達手段により、管理サーバと従属サーバの間で、アプリケーションの実行可能処理の同期をとり各サーバでオペレータの操作により、処理アプリケーションの実行要求を行なう方法を取らざるを得ないという問題点がある。

【0008】本発明では、広範囲のネットワーク環境に接続されたクライアントサーバ形態のシステムで管理サーバと従属サーバのオンラインやバッチまたは非稼働といった各サーバでのアプリケーションの実行処理ステータスを管理サーバと従属サーバで個別に保有するのではなく、管理サーバで一元管理することに加え、管理サーバで保有管理している各サーバのアプリケーションの実行処理ステータスと、予め設定値として定義されたアプリケーションの実行条件に従い、各従属サーバでのオペレータ操作を必要とせず、管理サーバのみのオペレータ操作により一括して、各従属サーバにアプリケーション処理のアプリケーションの実行要求を送信することにより、各従属サーバのアプリケーション処理をアプリケーションの実行することを実現することを目的とする。

#### 【0009】

【課題を解決するための手段】本発明では、CSS形態で稼働しているシステム環境下で、複数台のサーバのうち1台の管理サーバとその他のサーバである従属サーバ内で複数のプロセスを生成し処理を実行する機能と、時刻や個々のサーバ内で動作するプロセスで出力されるメッセージをトリガーとしてプロセスを生成し処理を実行する機能を用いる。これにより、予め設定された条件による管理サーバと従属サーバ内のオンラインやバッチ等のサーバ内アプリケーションの実行処理ステータスのチェックを行うステータス制御部とアプリケーション実行要求に対して実行可否を判定し、当該処理の実行制御を行う実行制御部をそれぞれ独立したプロセスとして生成し処理を実行することを可能とすると同時に、それぞれの機能を相互に出力されるメッセージをトリガーとして予め設定された実行条件に基づいて処理の実行制御を行うことを可能としている。まず、管理サーバのステータス制御部は、管理サーバ内で自管理サーバとその配下にある従属サーバに対応した監視用のプロセスを、それぞれ別プロセスとして起動する。また、従属サーバのステータス制御部では管理サーバへのステータス通知を行うためのプロセスを管理サーバからの起動通知メッセージをトリガーとして起動する。これにより、管理サーバのステータス制御部と従属サーバのステータス制御部との間で、予め決められた間隔でサーバの生死監視用のメッセージを送受信することにより、管理サーバと従属サーバ毎にそれぞれのサーバの生死監視を行うと同時に、管理サーバでの自サーバと個々の従属サーバの稼動状態を管理サーバと従属サーバとの間で必要情報を伝達するための連絡ファイルに保有し各サーバの稼動状態の一元管理を可能としている。また、実行制御部は、常時起動

されているものではなく、予め設定された起動時刻または、他プロセスで発生するメッセージをトリガーとして起動される。起動後は、連絡ファイルに保有している各サーバのステータスを参照し、それぞれのステータスに応じた処理を予め設定された実行条件に基づいて実行するため、それぞれの処理要求に対応したプロセスを生成することにより、各サーバ内の実行可能処理の制御を行う。これと同時に、前記連絡ファイル内に保有する自サーバのアプリケーション実行状態のステータスを変更する。

【0010】さらに、管理サーバと従属サーバの実行制御部では、管理サーバからの従属サーバへの実行指示と、それに対する従属サーバから管理サーバへの処理起動完了通知をメッセージとして送受信することにより、管理サーバ内の連絡ファイルで保有する管理サーバと個々の従属サーバのアプリケーション実行状態のステータスを変更し、全サーバのアプリケーション実行状態の一元管理を可能としていると同時に、管理サーバからの従属サーバに対する実行可能処理の一括制御を可能としている。

【0011】結果として、管理サーバでは、連絡ファイルを利用することにより、自サーバと全従属サーバの稼動状態とアプリケーション実行状態の一元管理することが可能となると同時に、予め定義された内容と自サーバと各従属サーバ毎のアプリケーション実行状態を参照することにより実行可能処理を判断し、自サーバと従属サーバに応じた処理の実行を一括制御することが可能となる。

【0012】また、ステータス制御部と実行制御部を管理サーバでは個々の従属サーバに対応して別々のプロセスとして起動しているため、管理サーバでは従属サーバ毎の稼動状態の管理や処理実行指示による実行処理の制御が可能である。従属サーバでも他の従属サーバの稼動状態やアプリケーションの実行状態に関係なく、管理サーバからの処理実行指示に応じた個々のサーバに必要とされる処理の実行が可能である。

#### 【0013】

【発明の実施の形態】以下、図面を用いて、本発明の一実施例を説明する。

【0014】図1は、本発明によるCSSサーバの一元管理および処理実行制御の仕組みの概念図を示したものである。

【0015】100は管理サーバ、200は管理サーバ100によって、運用を制御される従属サーバであり、それぞれのサーバは、ネットワーク300を介して接続される。

【0016】各サーバ内には、ステータス制御部（実行プロセス）110/210、実行制御部（実行プロセス）120/220、連絡ファイル130/230、メッセージ送受信プログラム140/240、その他のプロセス150/250が存在する。

【0017】管理サーバ100および従属サーバ200のステ

ータス制御部110/210は、それぞれのサーバの稼働状態を連絡ファイル130/230に書き込んでいる(510)。

【0018】管理サーバ100内のステータス制御部110は、従属サーバ200の数だけ存在している。それぞれが従属サーバ200内のステータス制御部210に対して、メッセージ送受信プログラム140/240を利用して、一定間隔でメッセージを送信している(520)。従属サーバ200のステータス制御部210は、その応答として、メッセージ送受信プログラム140/240を利用して、管理サーバ100内のステータス制御部110に対し、メッセージを送信する(530)。

【0019】管理サーバ100において、アプリケーション起動等の処理要求が入力される(540)と、実行制御部120が、連絡ファイル130内の自サーバの状態を読み込み(550)、処理要求を受け付け可能かチェックする。処理要求が受け付け可能であれば、連絡ファイル130内の自サーバステータスを変更し(560)、処理要求に見合ったプロセス150を起動する(570)。

【0020】さらに、実行制御部120は、連絡ファイル130内の従属サーバ200の状態を読み込み(550)、当該処理要求を行う従属サーバを特定する。そして、メッセージ送受信プログラム140/240を利用して、従属サーバ200内の実行制御部220に対し、処理の実行を指示する(580)。一方、従属サーバ200内の実行制御部220は、連絡ファイル230内の自サーバステータスを読み込み(590)、処理要求を受け付け可能かチェックする。処理要求が受け付け可能であれば、連絡ファイル230内の自サーバステータスを変更し(600)、処理要求に見合ったプロセス250を起動する(610)。そして、プロセスの正常起動後に、管理サーバ100の実行制御部120に対し、メッセージ送受信プログラム140/240を介して、起動完了を通知する(620)。実行制御部120では、これを受け、連絡ファイル130内の従属サーバ200のステータスを変更する(560)。

【0021】図2は、連絡ファイル130の構成例であり、ホスト名称を示すフィールド131、サーバが停止中か稼働中かといった稼働状態を示すフィールド132、アプリケーションの実行状態を示すフィールド133から成る。また、連絡ファイル230については、1つの従属サーバについて、同様のフィールドから構成される。

【0022】図3、4、5は、本発明におけるステータス制御部(処理110と処理210)実施例のフローチャートであり、図6は同様に実行制御部(処理120と処理220)の実施例のフローチャートである。また、管理サーバと各従属サーバのステータス制御部で送受信するメッセージのフォーマットを図7に、管理サーバと各従属サーバの実行制御部で送受信する実行指示・起動完了通知メッセージのフォーマットを図8に示す。

【0023】本処理を開始する場合、まずステータス制御部(処理110と処理210)を管理サーバと従属サーバ起動時に開始する。(処理10~11)

ステータス制御部は起動されると、起動されたサーバが管理サーバか従属サーバかを予め定義されたサーバ定義情報に基づいて判断する。(処理12)

管理サーバの場合は、各従属サーバのサーバ稼働状態ステータスを監視するための監視プロセスの起動処理を実行する。(処理13) まず、自サーバと各従属サーバのサーバ稼働状態ステータスとアプリケーションサーバ稼働状態を監視・保有するための連絡ファイルを(ファイル130)作成する。(処理14) そして、管理サーバ起動終了後、自サーバと各従属サーバのステータスを監視するための監視プロセスを自サーバと予め定義している配下に接続されている従属サーバの数分起動する。(処理15) 起動された監視プロセスは、予め設定された自サーバと各従属サーバのホスト名称・稼働状態監視間隔・エラー発生時のリトライ回数・応答待ち時間の定義データを読み込みステータス監視の制御データの初期設定を行う。(処理16) ここで初期設定された制御データをもとに、各従属サーバに対して、サーバの稼働状態を確認するための図7に示すフォーマットの管理サーバ送信メッセージを各従属サーバに対して送信する。(処理18) このメッセージ送信前に、管理サーバではサーバ稼働状態ステータスを更新するための管理プロセスを起動する。(処理17) さらに、監視プロセスは、監視プロセスプログラム内部で保有しているテーブル内のメッセージ送信終了フラグを「ON」にし、制御データに設定した応答待ち時間が経過するまでの間、各従属サーバからの図7に示すフォーマットの管理サーバ受信メッセージの受信待ちをする。(処理19) 応答待ち時間内に各従属サーバからのサーバ稼働中の応答メッセージ受信した場合(フィールド86が稼働中のビット)、監視プロセスは、先の送信終了フラグを「OFF」にする。また、この時管理プロセスに対してサーバ稼働状態ステータスの更新要求を発行し、管理プロセスではその従属サーバは稼働状態と判定し、サーバ稼働状態ステータス(フィールド132)を「稼働中」にする。(処理21) この一連の処理(処理17~21)を制御データに設定した稼働状態監視間隔で管理サーバと各従属サーバの間でメッセージの送受信を行うことにより、管理サーバでの各従属サーバの稼働状態ステータスの一元管理を継続実施する。

【0024】また、管理サーバは、各従属サーバからの管理サーバ受信メッセージを応答待ち時間内に受信できなかった場合、監視プロセスは制御ファイルに設定したリトライ回数分、管理サーバと従属サーバの間でメッセージの送受信を実行し(処理22)、その間1度も管理サーバで従属サーバからの応答メッセージを受信出来ない場合(無応答状態)は、その従属サーバは停止状態と判断して、管理プロセスはサーバ稼働状態ステータス(フィールド132)を「停止中」にする。(処理19、23) また、従属サーバからの管理サーバ受信メッセージがサーバ停止を表すメッセージの場合(フィールド86が停止中のビッ

ト)も、管理プロセスはサーバ稼動状態ステータス(フィールド132)を「停止中」にする。そして、管理サーバからのサーバ稼動状態ステータス(フィールド132)が「停止中」の従属サーバに対する先に示した管理プロセスを終了する。(処理24)しかし、監視プロセスは先に示した処理17~20の処理を繰り返しており、この間に従属サーバから管理サーバに対して、サーバ稼動中の管理サーバ受信メッセージを再受信した場合は、処理17に示す管理プロセスの起動を行う。(処理28~30)管理プロセスはサーバ稼動状態ステータス(フィールド132)を「稼動中」にし、管理サーバからの稼動状態監視を継続する。

【0025】また、管理サーバではオペレータからの監視終了要求または、管理サーバのシステム終了処理(システム停止処理)時に、ステータス制御部の終了処理を実行し、管理プロセスを停止し、連絡ファイル(ファイル130)をクリアし、その後起動している監視プロセスを終了する。(処理25~27)

一方、各従属サーバでは、管理サーバと同様に監視プロセスを起動する。(処理31)また監視プロセスは、起動処理中に管理サーバと同様のフォーマットを持つ連絡ファイル(ファイル230)を作成する。そして、自サーバのサーバ稼動状態を更新するための管理プロセスを起動する。(処理32)従属サーバの監視プロセスでは、管理サーバから一定間隔で送信されるメッセージを受信し、それに対する応答メッセージを管理サーバに向けて送信する。(処理33、37)また、送信時に管理サーバと同様に管理プロセスに対して自サーバの稼動状態更新要求を行い、連絡ファイル(フォーマット230)内のサーバ稼動状態ステータス(フィールド132)を「稼動中」にする。(処理38)この処理(処理33~34、37~38)を管理サーバからの監視終了メッセージ受信時または、オペレータによる監視終了コマンド投入まで繰り返し実行する。但し、管理サーバからの送信メッセージが監視終了メッセージの場合または、オペレータからの監視終了コマンド投入時は、監視プロセスは、管理プロセスを停止し連絡ファイル(ファイル230)をクリアする。その後、ステータス制御部は監視プロセスの終了処理を行う。(処理35~36)実行制御部(処理120、220)は、ステータス制御部のように常時起動されているものではなく、予め設定された時刻または、各サーバ内他プロセスで出力するメッセージをトリガーとして、起動される。(処理50、51)実行制御部も先に述べたステータス制御部と同様に、予め定義されたサーバ定義情報に基づいて、起動されたサーバが管理サーバか従属サーバかを判断する。(処理52)管理サーバの場合は、予め設定された自サーバと各従属サーバの実行処理定義ファイル(ファイル90)を実行制御部の内部情報として保持し、起動対象処理の確定を行う。(処理53)先に述べた時刻または他プロセスで出力するメッセージをトリガーとして起動されると、ステータス制御部で更新・管理している連絡ファイル(ファイル13

0)の内部に保有する、サーバ稼動状態ステータス(フィールド132)とアプリケーション実行状態ステータス(フィールド133)のチェック処理を実行する。(処理54)このチェック対象としている自サーバまたは従属サーバのそれについて、稼動状態が「稼動中」かつ実行しようとしているアプリケーション処理が未実行状態であれば、自サーバもしくは従属サーバに対して、図8に示すフォーマットの管理サーバ送信メッセージを用いて実行指示処理のID(フォーマット91、フォーマット90が起動要求のビット)を送信する。(処理55~56)自サーバまたは従属サーバの稼動状態が「停止中」もしくは実行しようとしているアプリケーション処理が既に実行済みである場合は、何も処理をせずに管理サーバの実行制御部の処理を終了する。(処理60)その後、自サーバ内の実行完了応答または、従属サーバからの図8に示すフォーマットの管理サーバ受信メッセージ送信されるで該当処理の起動完了通知(フォーマット94が起動完了のビット)の受信待ち状態に入る。(処理57)

実行指示を受けた従属サーバでは、その実行指示受信に従い、自サーバ内の実行制御部の起動を行う。(処理61)

実行制御部の起動終了後、従属サーバの実行制御部は該当する処理が未実行か否かを自サーバ内に保有している連絡ファイル(ファイル230)を参照し、再チェックを実行する。(処理63)この再チェックを行うことにより、管理サーバと従属サーバ間の伝送上の制御電文のずれ違いによる2重起動を防止している。チェックの結果、未実行であれば管理サーバからの実行指示を受けた処理を実行する。(処理64)その処理の正常終了を受けて、自サーバ内の連絡ファイル(ファイル230)のアプリケーション実行状態ステータス(フィールド133)を更新する。(処理65)実行指示を受けた処理が正常終了しなかった場合は、アプリケーション実行状態ステータス(フィールド133)の更新は実行しない。またチェックの結果、実行指示を受けた処理が既に起動されている状態にある場合は、実行指示を受けた処理の実行は行わず、アプリケーション実行状態ステータス(フィールド133)の更新も実行しない。いずれの場合もアプリケーション実行状態ステータス(フィールド133)に対する処理が終了した時点で、管理サーバ受信メッセージ(図6)で起動完了の旨を管理サーバ側へ通知する。(処理66)従属サーバの実行制御部の処理は、この起動完了通知のメッセージの送信完了後終了する。(処理67)

管理サーバ側ではこの管理サーバ受信メッセージを受信し、メッセージの待ち状態を解除する。(処理62)受信した管理サーバ受信メッセージの内容が、実行指示した処理が正常終了(フォーマット94が起動完了のビット)した場合は、管理サーバ内の連絡ファイル(ファイル130)のアプリケーション実行状態ステータス(フィールド133)を更新を行い、管理サーバの実行制御部の処理を終了する。(処理59~60)実行指示した処理が正常終了し

なかった場合は、正常終了しなかった旨のエラーメッセージを出し、アプリケーション実行状態ステータス(フィールド133)を更新せず、管理サーバの実行制御部の処理を終了する。(処理60)また、実行指示した処理が既に実行されていた場合は、管理サーバ受信メッセージの受信だけを行い、アプリケーション実行状態ステータス(フィールド133)を更新せず、管理サーバの実行制御部の処理を終了する。また、管理サーバ自体の実行指示に対する処理も、従属サーバでの処理と同様に実行指示した処理の正常終了時は、自サーバのアプリケーション実行状態ステータス(フィールド133)を更新し、管理サーバの実行制御部の処理を終了する。実行指示した処理が正常終了しなかった場合は、正常終了しなかった旨のメッセージを出し、アプリケーション実行状態ステータス(フィールド133)を更新せず、管理サーバの実行制御部の処理を終了する。

## 【0026】

【発明の効果】本発明の実施例によれば、次のような効果がある。

【0027】(1)大規模なCSS形態の分散系システムにおいて、1台の管理サーバでネットワーク上に接続される従属サーバ全ての稼動状態を生死監視ならびにアプリケーション実行状態として一元管理できる。

【0028】(2)個々のサーバに関する障害を迅速に把握することが可能となる。

【0029】(3)管理サーバで自サーバも含め個々の従属サーバの実行可能処理を予め設定された条件により実行可否を判断し、それに基づき管理・従属サーバではアプリケーションを実行することが可能なため、オペレーションレスでのアプリケーションの実行制御が可能となる。

【0030】(4)管理サーバでのステータス制御部・実行制御部は、各従属サーバ毎に別プロセスとして起動されることから、個々の従属サーバ個別の稼動状態監視・アプリケーション実行制御が可能となる。

## 【図面の簡単な説明】

【図1】CSSサーバの一元管理と処理実行制御の仕組みの概念図である。

【図2】連絡ファイル構成例である。

【図3】ステータス制御部のフローチャート例である。

【図4】ステータス制御部のフローチャート例である。

【図5】ステータス制御部のフローチャート例である。

【図6】実行制御部のフローチャート例である。

【図7】ステータス制御部の送信メッセージフォーマット例である。

【図8】実行制御部の送受信メッセージフォーマット例である。

## 【符号の説明】

100…管理サーバ、	200…従属サーバ、
110、210…ステータス制御部、	120、
220…実行制御部、	130、230…連絡ファイル、
150、250…その他のプロセス、	300…ネットワーク、
510…書き込み、	520…送信、
530…応答、	540…処理要求、
590…読み込み、	550、600…変更、
570、610…起動、	580…実行指示、
620…起動完了通知、	131…ホスト名、
32…サーバ稼働状態、	133…アプリケーション実行状態、
133…アプリケーション実行状態、	81、84、88、92…従属サーバホスト名、
82、85、89、93…電文種別、	83、86、90、94…制御コード、
87、96…付加メッセージ、	91、95…実行指示処理ID。

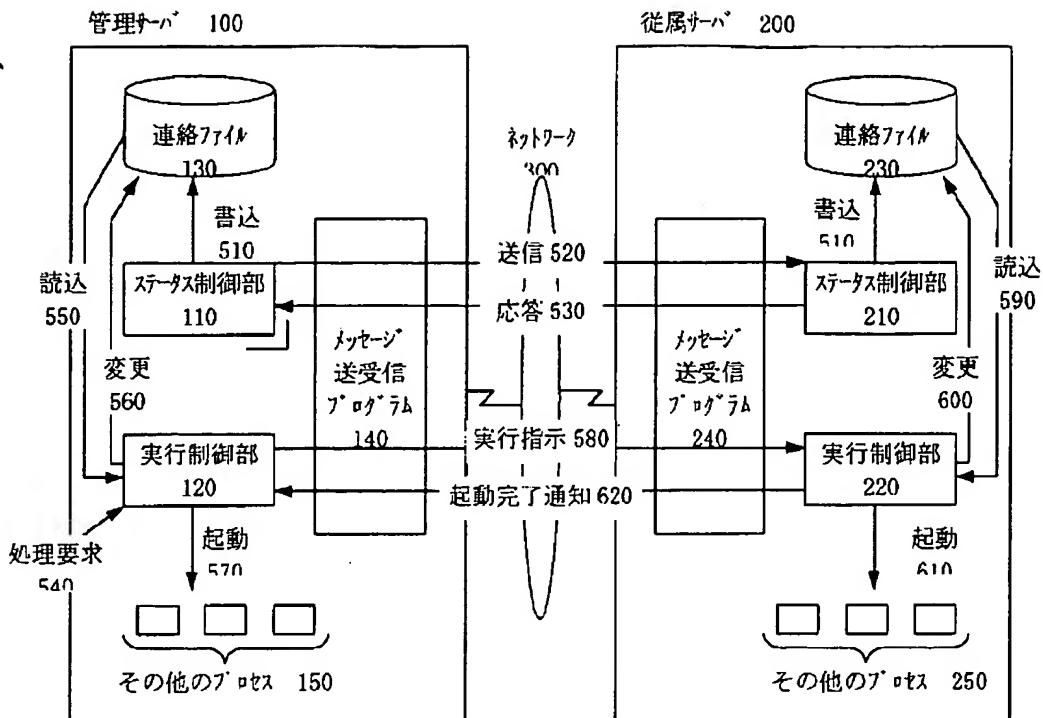
【図2】

図 2

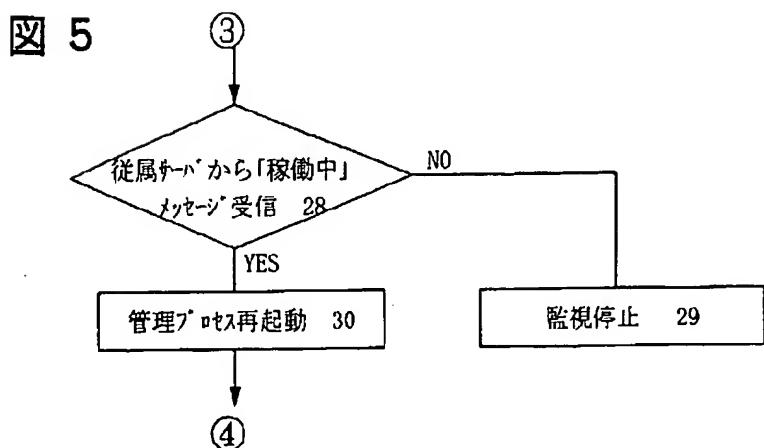
ホスト名 131	サーバ稼動状態 132	アプリケーション実行状態 133
管理サーバホスト名	稼働中	オンライン中
従属サーバホスト名-1	停止中	未起動
従属サーバホスト名-2	稼働中	バッチ中
従属サーバホスト名-2	稼働中	オンライン中

【図1】

図 1

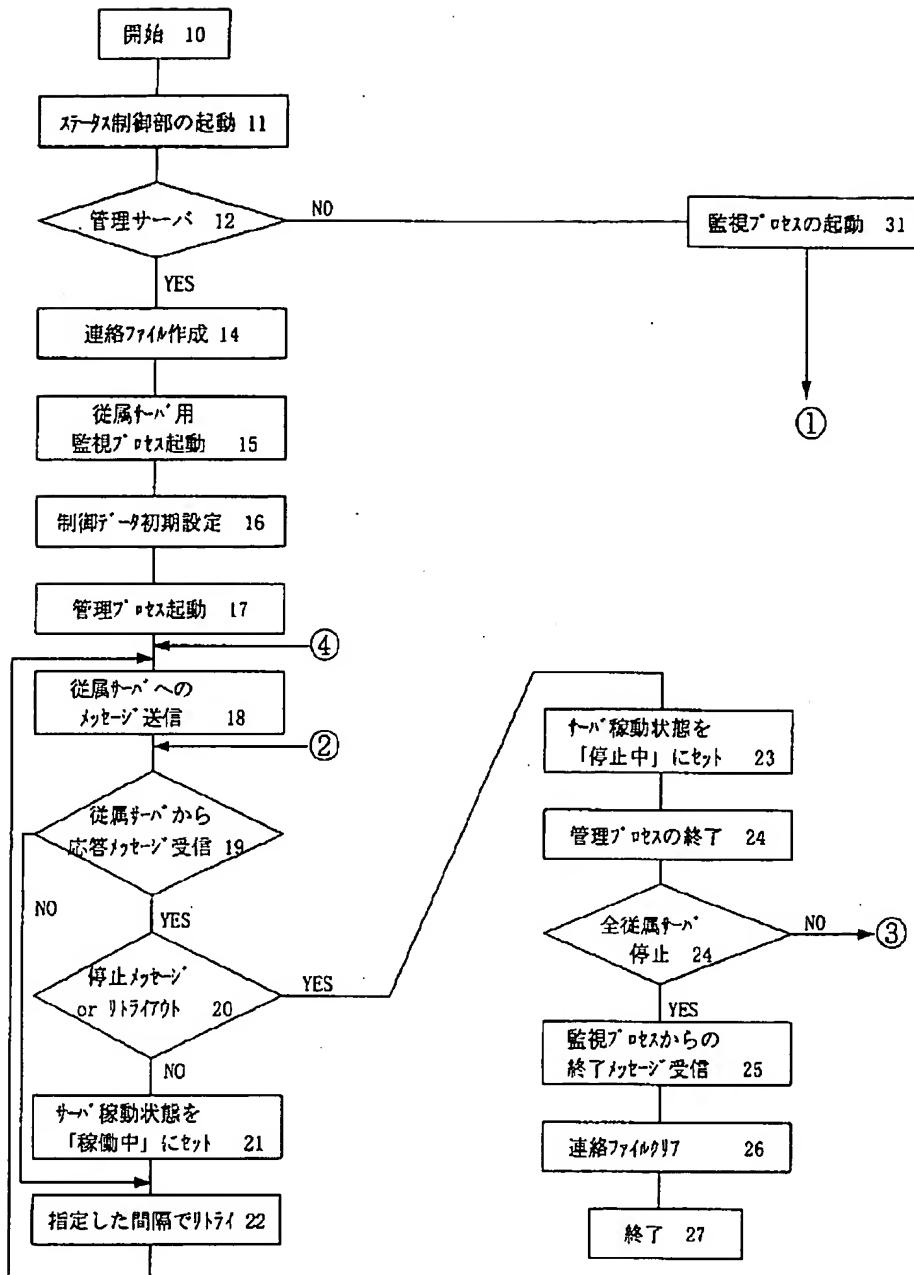


【図5】



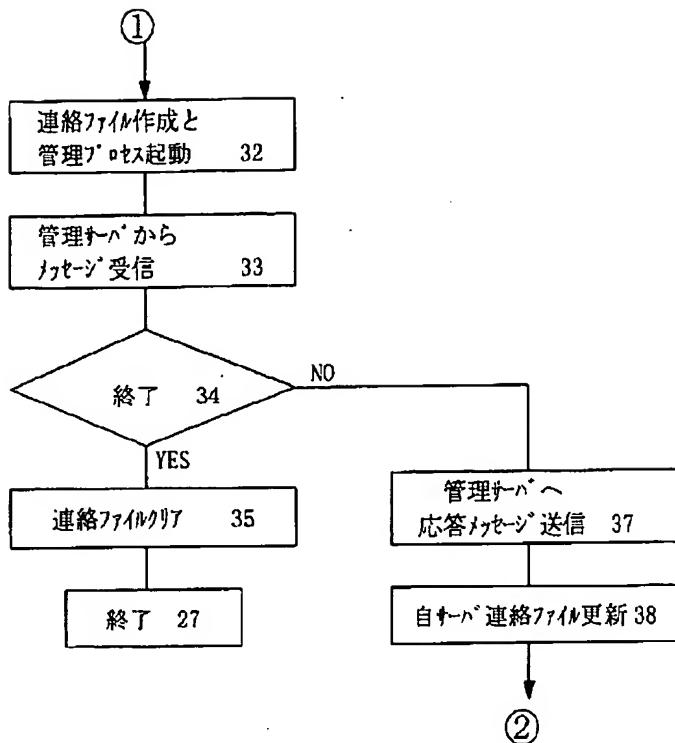
【 3】

3



【図4】

図 4



【図7】

図 7

管理サーバ送信メッセージ

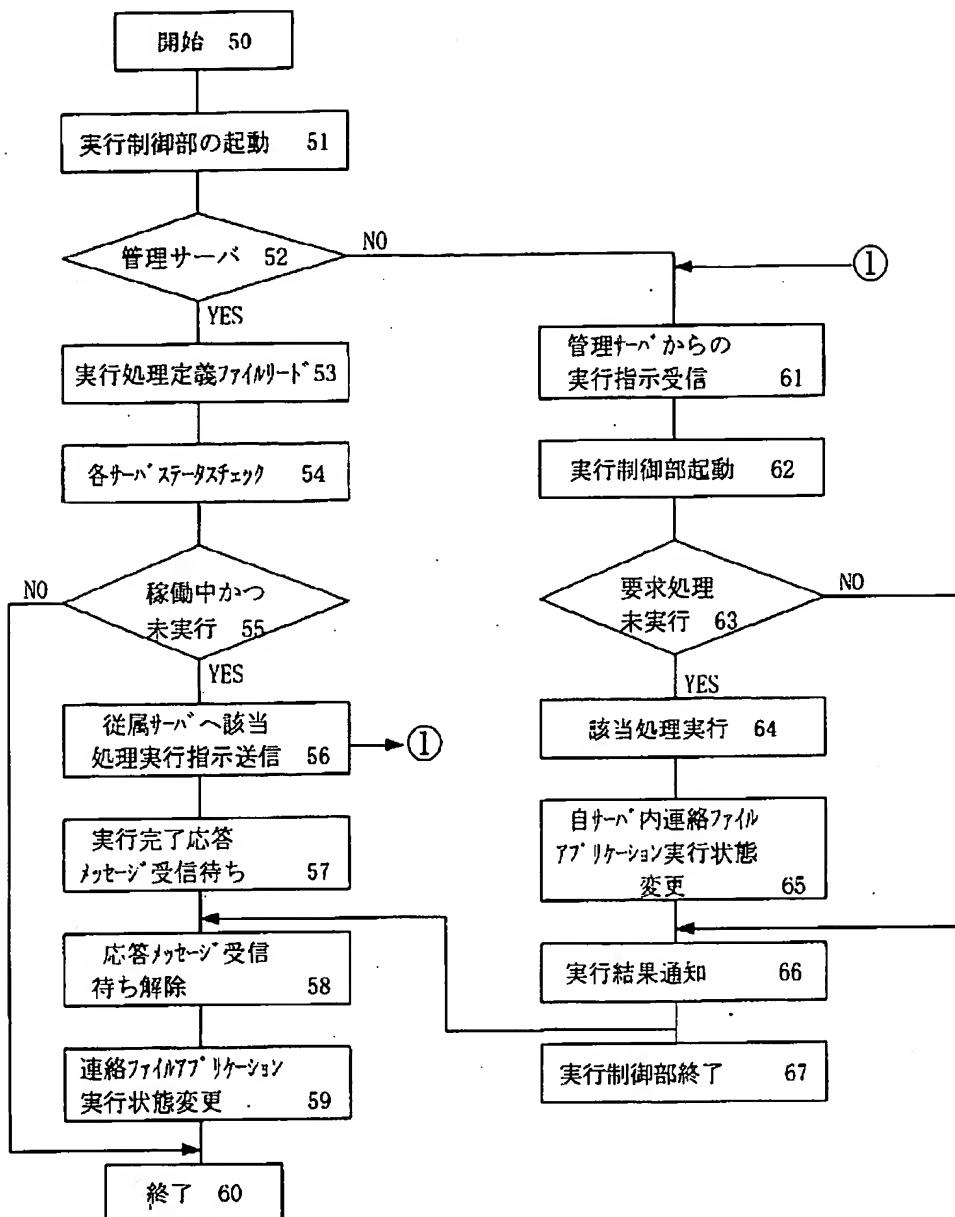
従属サーバホスト名 81	電文種別 82	制御コード 83
--------------	---------	----------

管理サーバ受信メッセージ

従属サーバホスト名 84	電文種別 85	制御コード 86	付加メッセージ 87
--------------	---------	----------	------------

【図6】

図 6



【図8】

## 図 8

## 管理サーバ送信メッセージ

従属サーバホスト名 88	電文種別 89	制御コード 90	実行指示処理 ID 91
--------------	---------	----------	--------------

## 管理サーバ受信メッセージ

従属サーバ ホスト名 92	電文種別 93	制御コード 94	実行指示処理 ID 91	付加メッセージ 87
------------------	---------	----------	-----------------	---------------